OIPE
JUN 27 2005
PATENT & TRADEMARK OFFICE

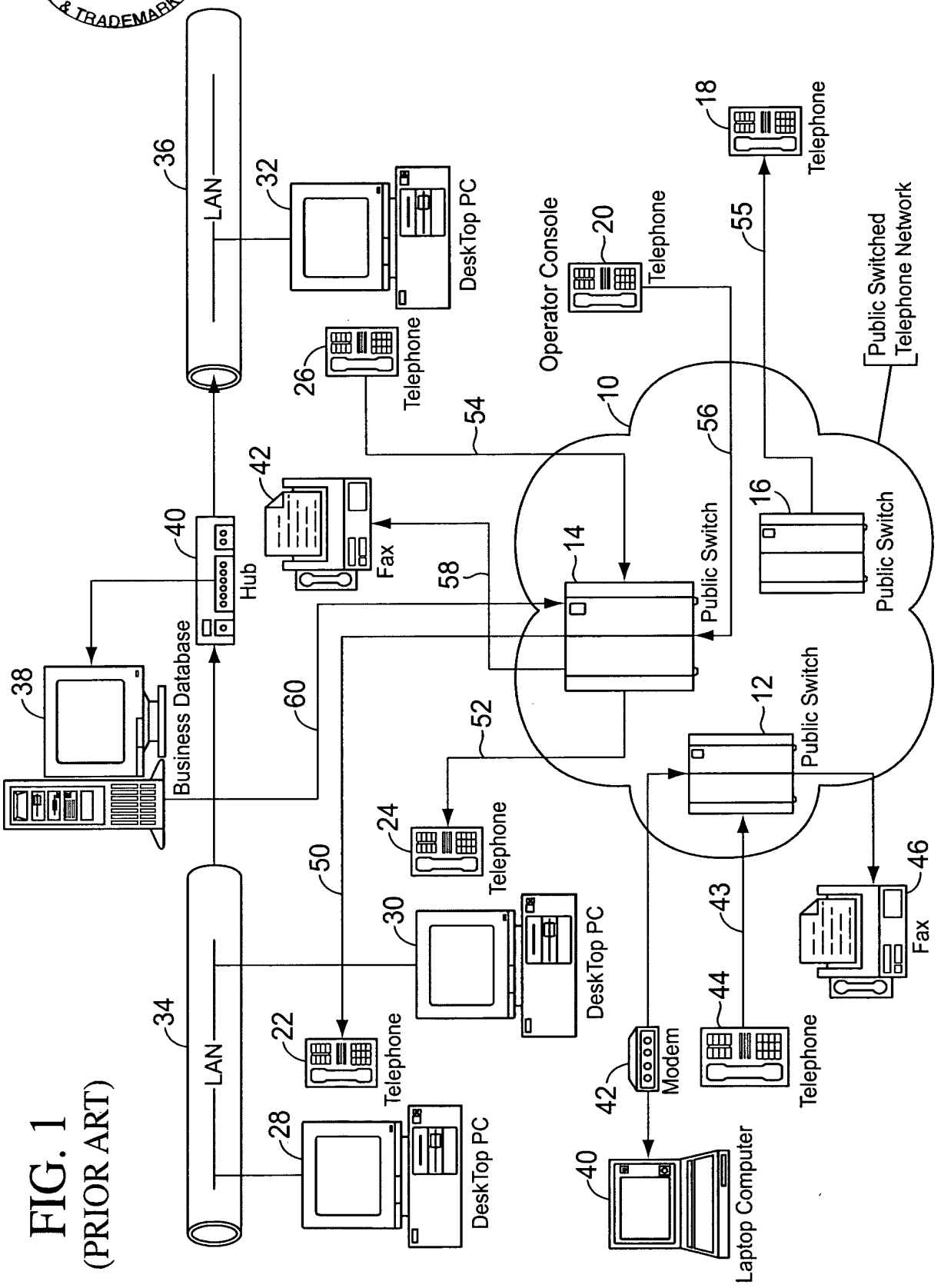


FIG. 1
(PRIOR ART)

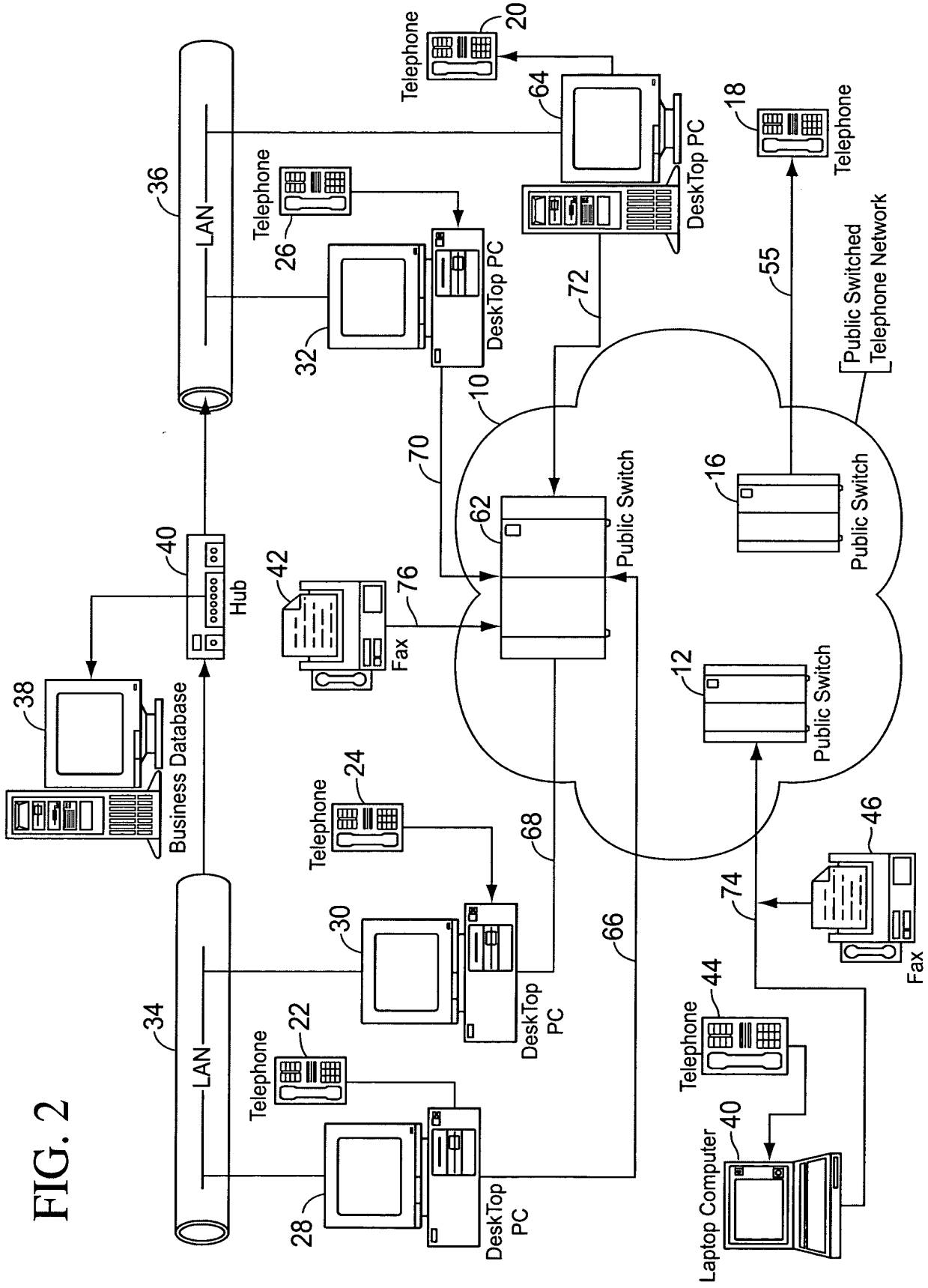

Telephone
20
64
DeskTop PC
18
Telephone
36
LAN
Telephone
26
DeskTop PC
72
55
Public Switched
Telephone Network
32
10
70
Public Switch
16
62
Public Switch
40
Hub
42
76
Fax
38
Business Database
12
Public Switch
Telephone
24
68
46
74
30
66
DeskTop
PC
Fax
34
LAN
Telephone
22
Telephone
44
28
DeskTop
PC
Laptop Computer
40

FIG. 2

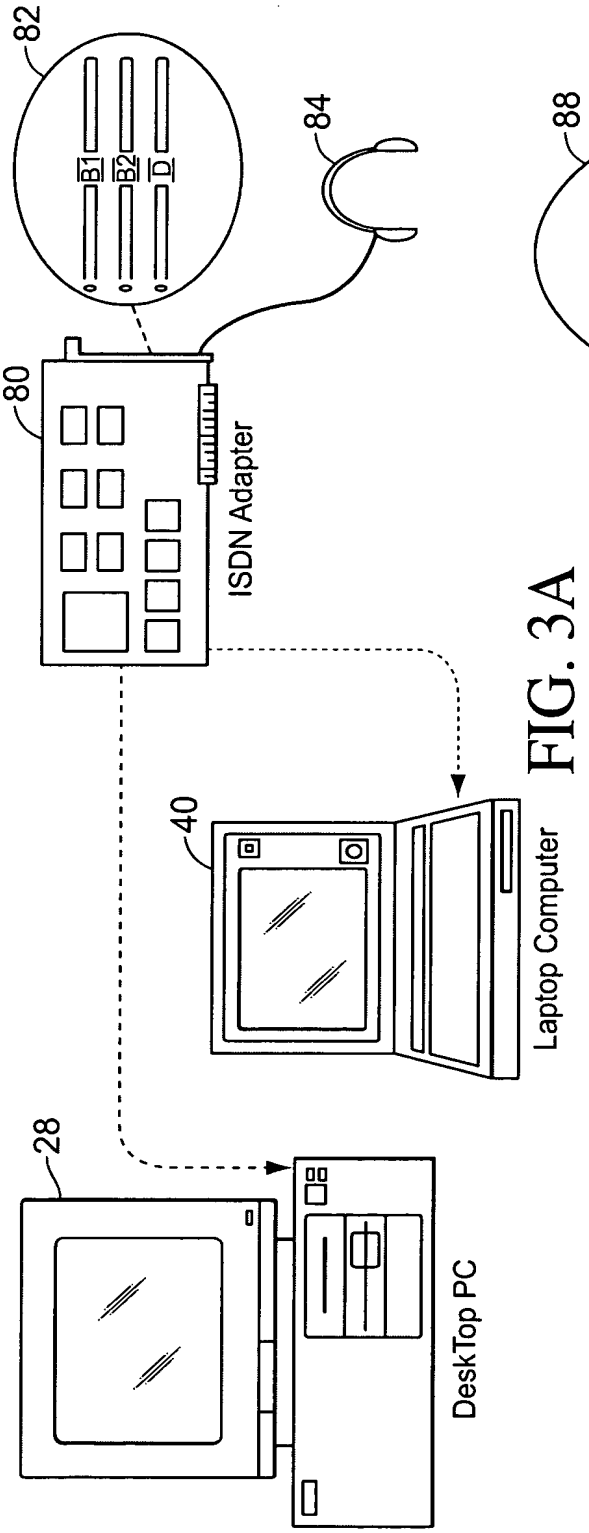

82
B1
B2
D
84
80
ISDN Adapter
40
Laptop Computer
28
DeskTop PC

FIG. 3A

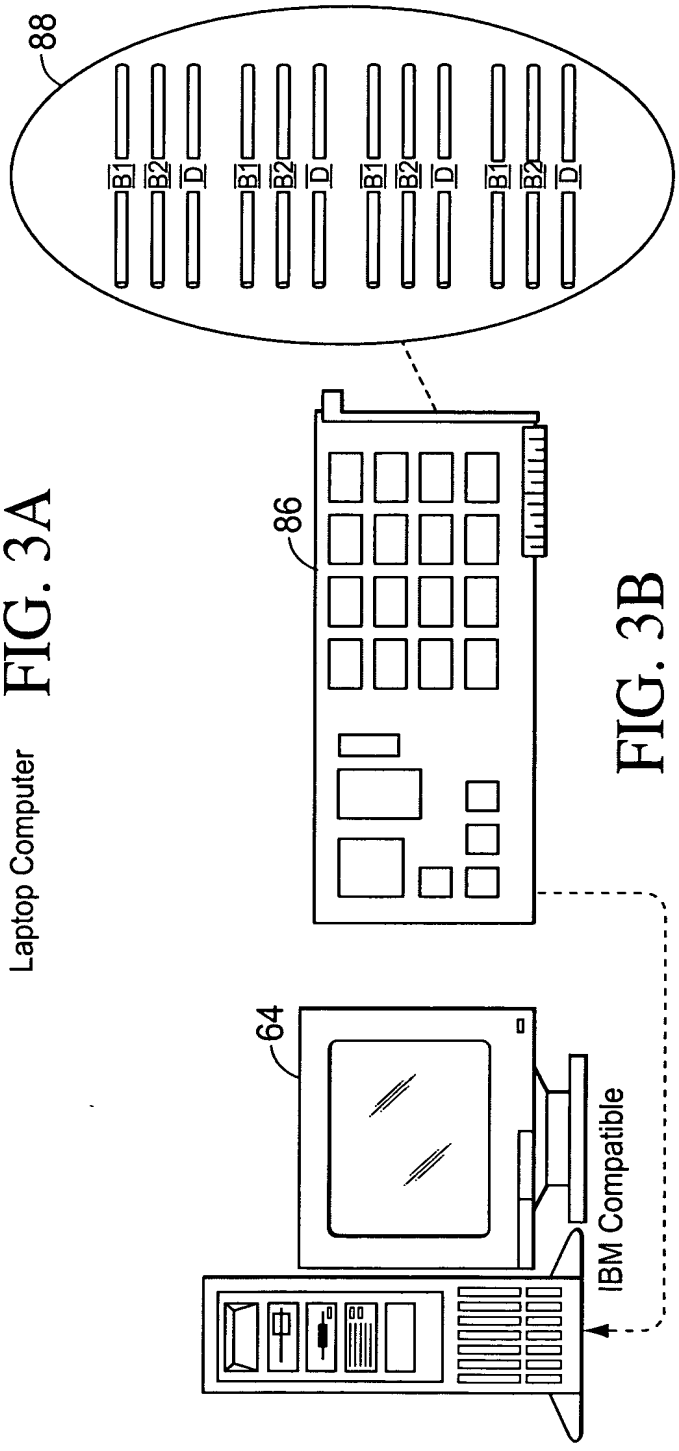

88
B1
B2
D
86
64
IBM Compatible

FIG. 3B

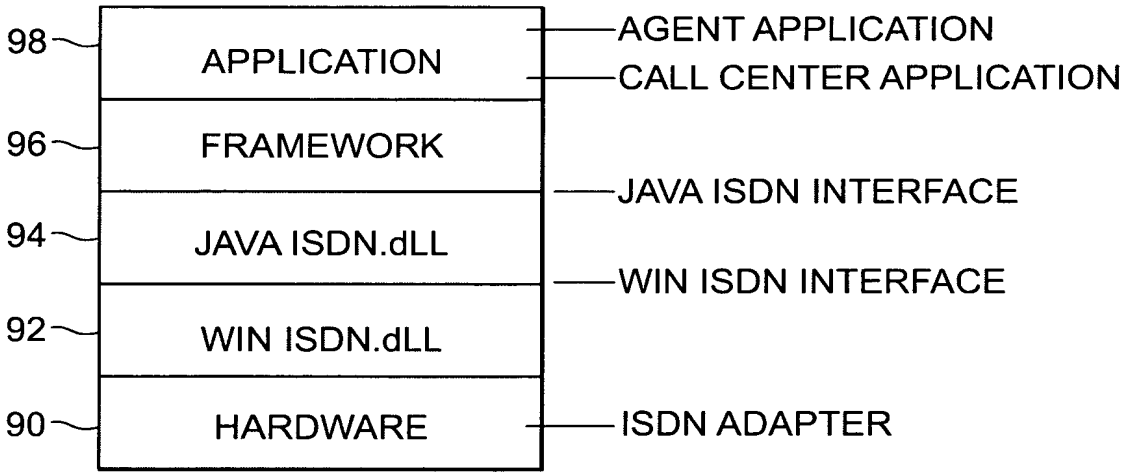

98
APPLICATION
96
FRAMEWORK
94
JAVA ISDN.dLL
92
WIN ISDN.dLL
90
HARDWARE
AGENT APPLICATION
CALL CENTER APPLICATION
JAVA ISDN INTERFACE
WIN ISDN INTERFACE
ISDN ADAPTER

FIG. 4

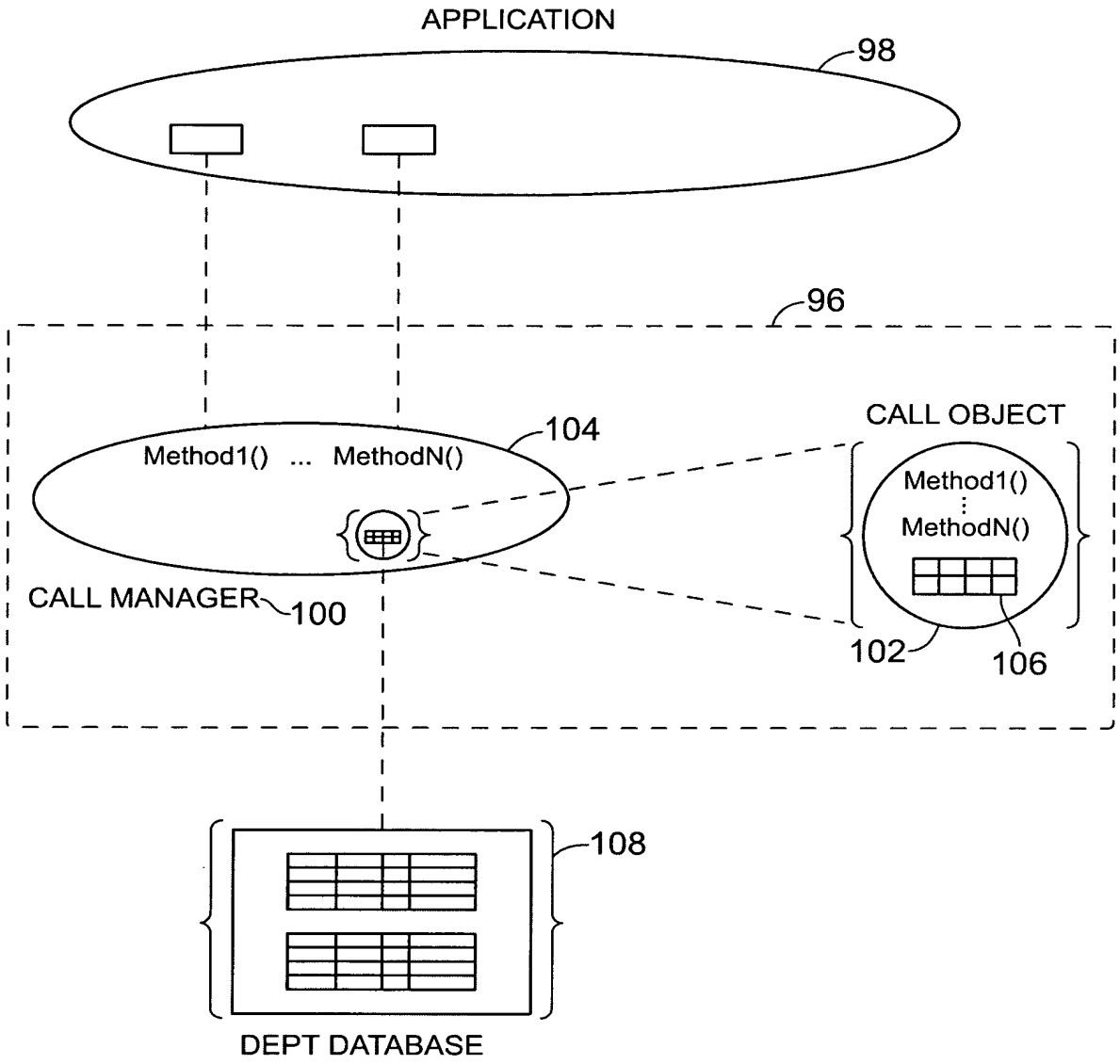

APPLICATION
98
96
104
Method1() ... MethodN()
CALL OBJECT
Method1()
MethodN()
CALL MANAGER
100
102
106
108
DEPT DATABASE

FIG. 5

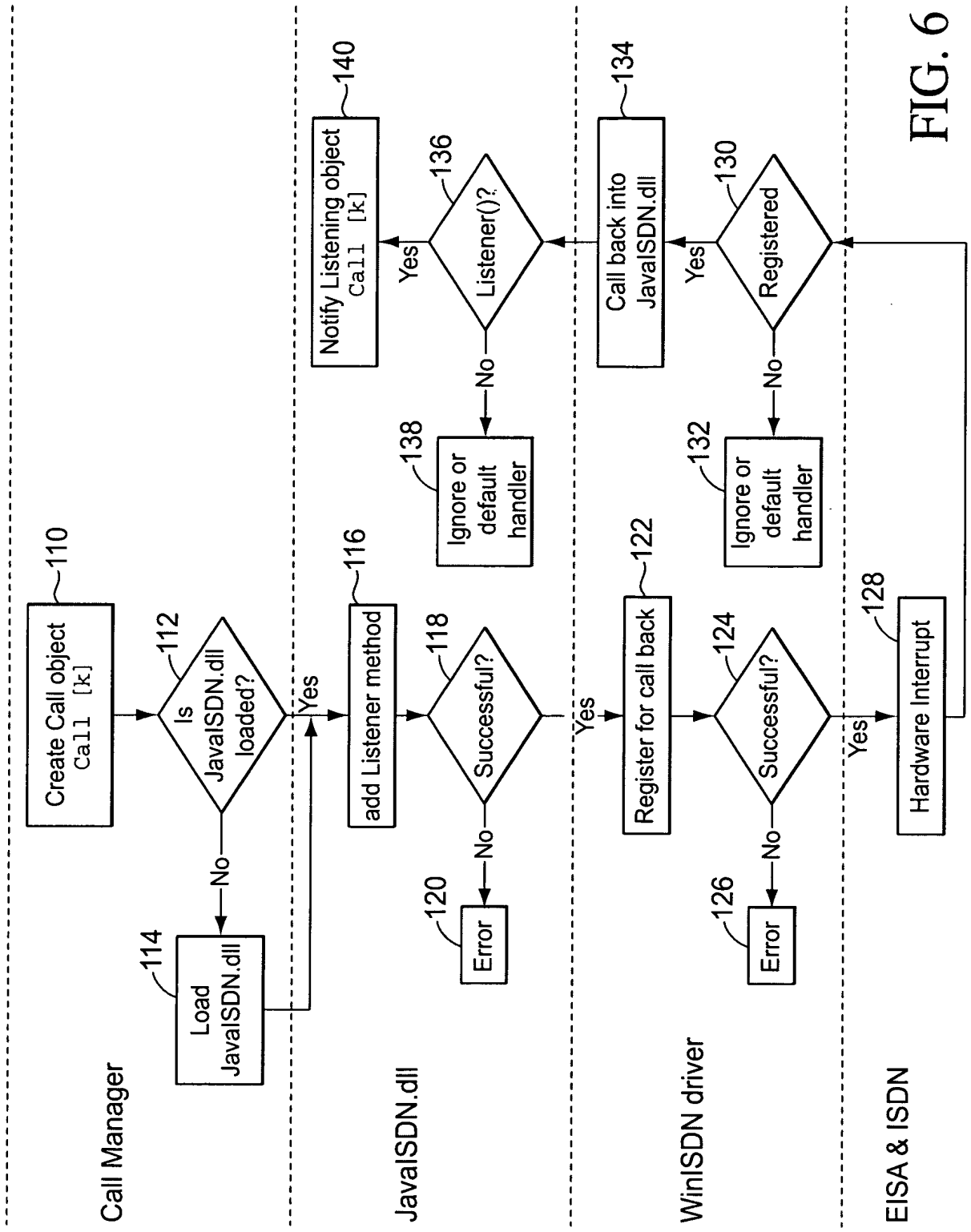

Call Manager
JavaISDN.dll
WinISDN driver
EISA & ISDN
Create Call object
Call [k]
110
Is JavaISDN.dll loaded?
112
No
Load JavaISDN.dll
114
Yes
add Listener method
116
Successful?
118
No
Error
120
Yes
Register for call back
122
Successful?
124
No
Error
126
Yes
Hardware Interrupt
128
Registered
130
No
Ignore or default handler
132
Yes
Call back into JavaISDN.dll
134
Listener()?
136
No
Ignore or default handler
138
Yes
Notify Listening object
Call [k]
140

FIG. 6

150 Menu Item	152 Next Menu	154 Next Dept	156 Voice Mail	158 Transfer to	160 Directory #	162 Agent Status
Oper		Oper		Oper	33	✓
Sales	test.spl	Sales				X
Tech	main.spl	Tech		Eddie	97671111	✓
Acct	Acct.au	Acct	test.spl			X
Ship			main.spl			X

TABLE 1

Menu Item	Next Menu	Next Dept	Voice Mail	Transfer to	Directory #	Agent Status
Domestic		Oper		Karen	41	X
International	main.spl	Main				X
OEM	test.spl	OEM1				✓
						X
						X

TABLE 2

FIG. 7

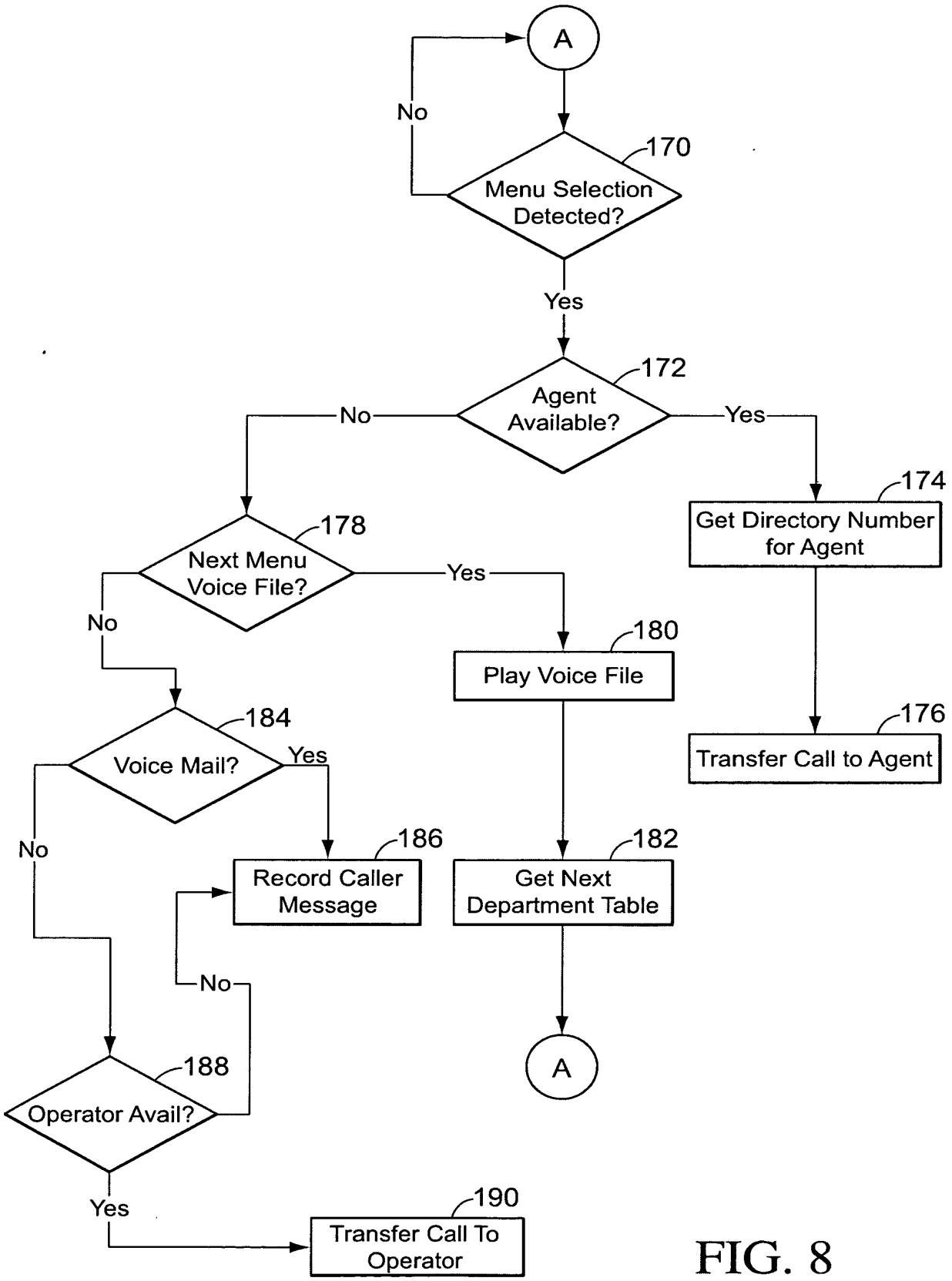

A
No
170
Menu Selection Detected?
Yes
172
Agent Available?
No
Yes
174
Get Directory Number for Agent
178
Next Menu Voice File?
Yes
No
180
Play Voice File
184
Voice Mail?
Yes
176
Transfer Call to Agent
No
186
Record Caller Message
182
Get Next Department Table
No
A
188
Operator Avail?
Yes
190
Transfer Call To Operator

FIG. 8

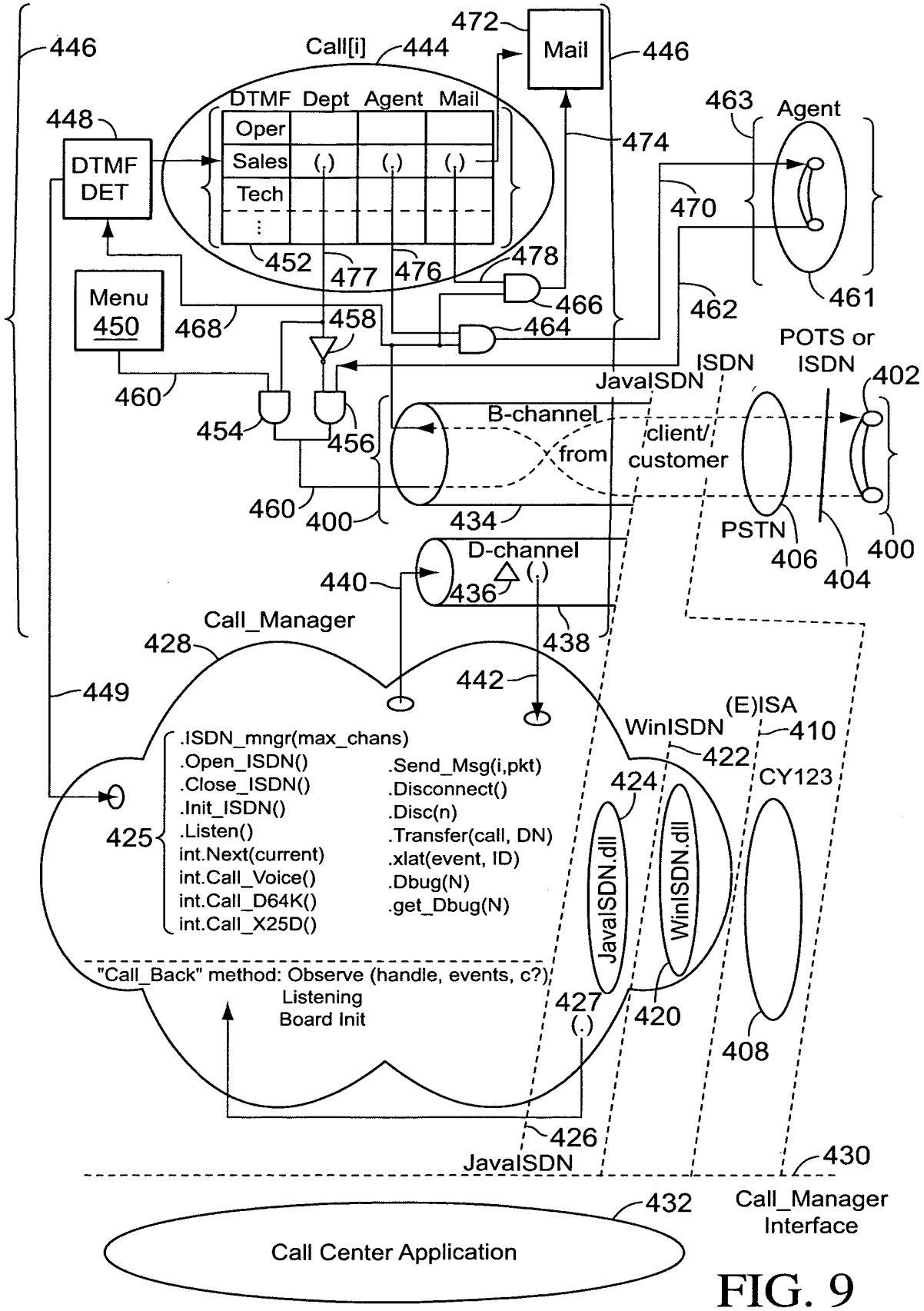

446
Call[i]
444
472
Mail
446
DTMF Dept Agent Mail
Oper
Sales
Tech
448
DTMF DET
463
Agent
474
470
452
477
476
478
466
462
461
Menu 450
468
458
464
POTS or ISDN
402
460
454
456
JavaISDN
ISDN
B-channel
client/ customer
from
460
400
434
PSTN
406
404
400
D-channel
440
436
438
Call_Manager
428
442
449
(E)ISA
WinISDN
410
422
424
CY123
.ISDN_mngr(max_chans)
.Open_ISDN()
.Close_ISDN()
.Init_ISDN()
.Listen()
int.Next(current)
int.Call_Voice()
int.Call_D64K()
int.Call_X25D()
.Send_Msg(i,pkt)
.Disconnect()
.Disc(n)
.Transfer(call, DN)
.xlat(event, ID)
.Dbug(N)
.get_Dbug(N)
425
JavaISDN.dll
WinISDN.dll
"Call_Back" method: Observe (handle, events, c?)
Listening
Board Init
427
420
408
426
JavaISDN
430
Call_Manager Interface
432
Call Center Application

FIG. 9

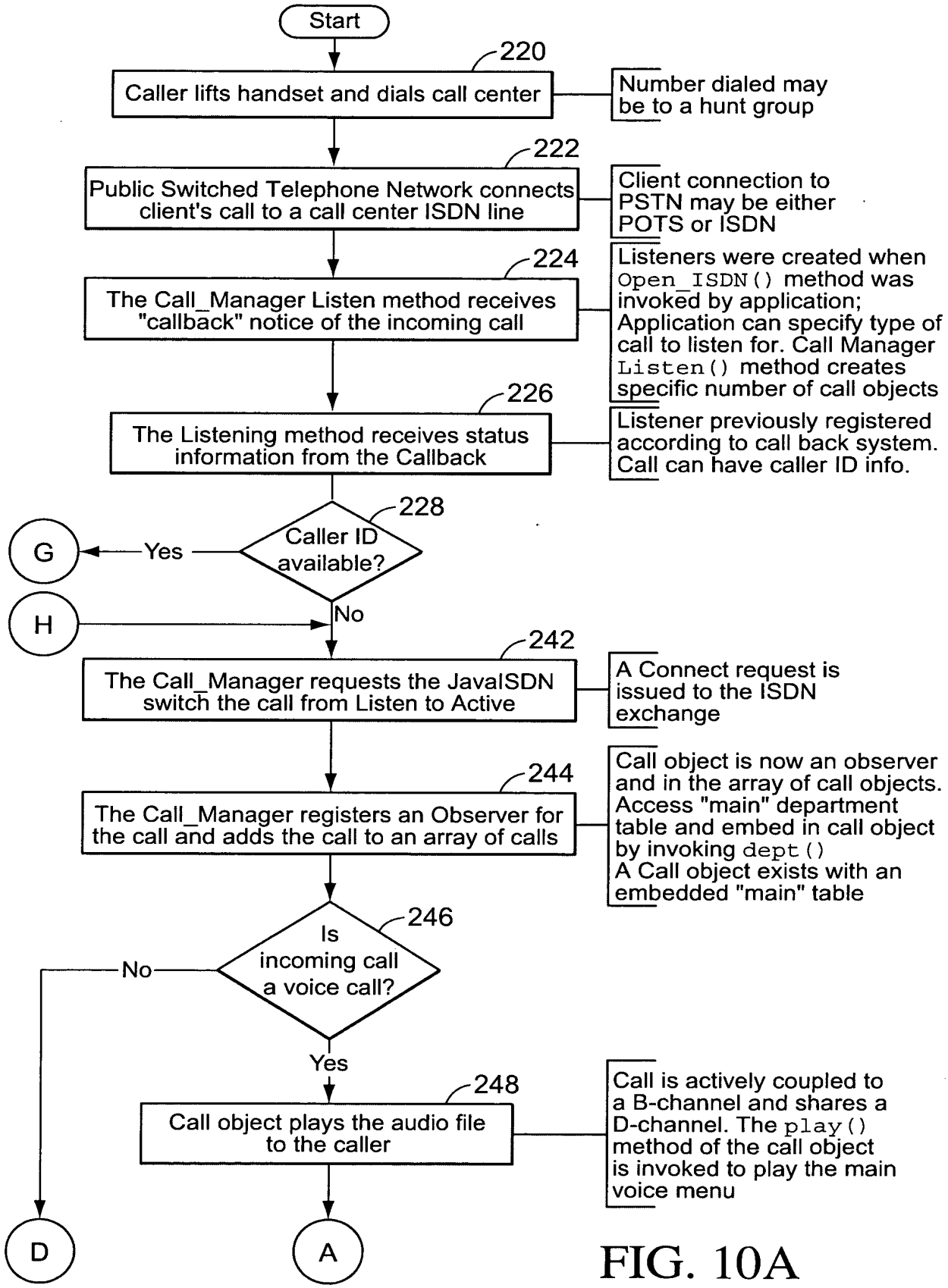
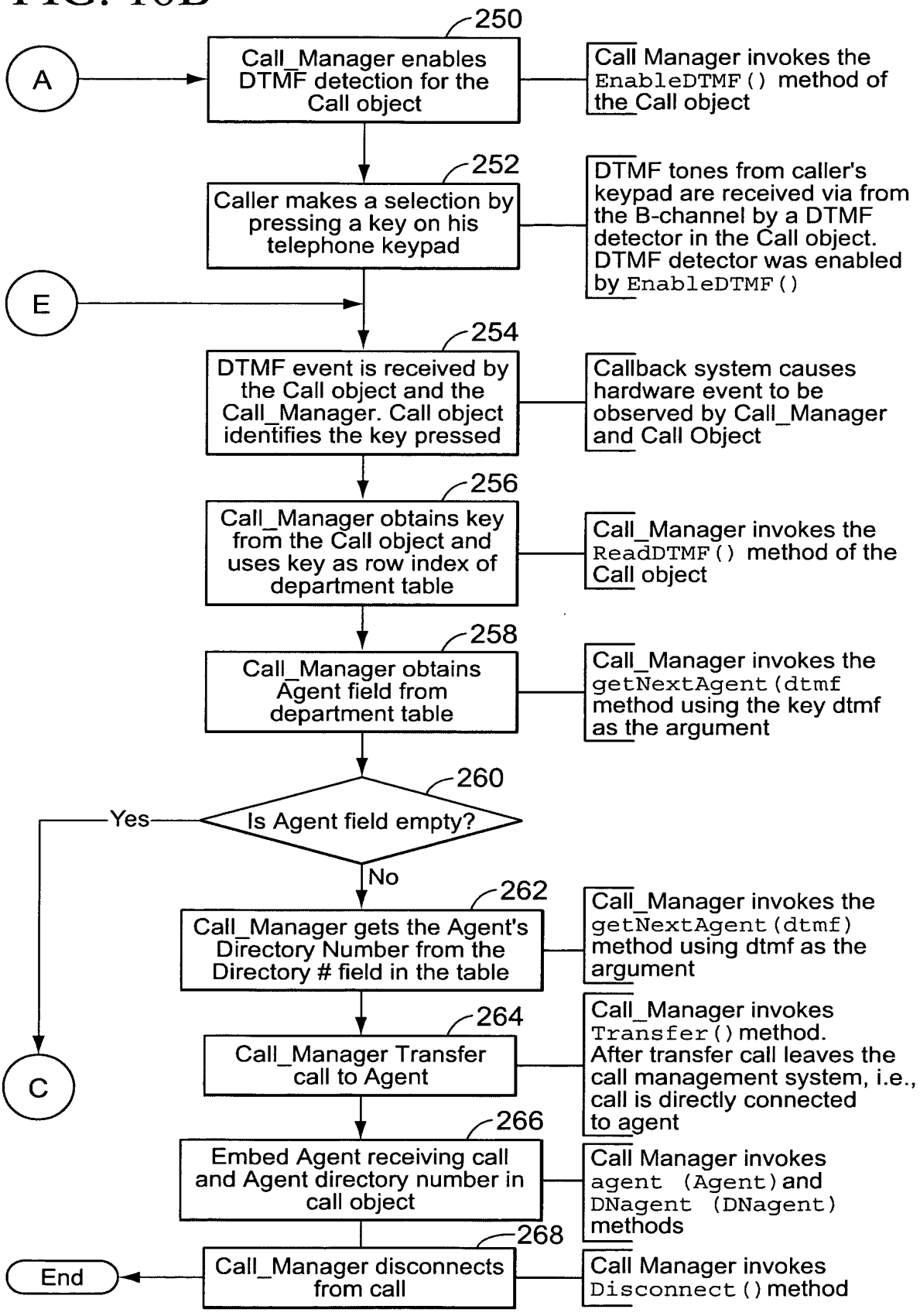

Start
220
Caller lifts handset and dials call center
Number dialed may be to a hunt group
222
Public Switched Telephone Network connects client's call to a call center ISDN line
Client connection to PSTN may be either POTS or ISDN
224
The Call_Manager Listen method receives "callback" notice of the incoming call
Listeners were created when Open_ISDN() method was invoked by application; Application can specify type of call to listen for. Call Manager Listen() method creates specific number of call objects
226
The Listening method receives status information from the Callback
Listener previously registered according to call back system. Call can have caller ID info.
228
Caller ID available?
Yes
G
No
H
242
The Call_Manager requests the JavaISDN switch the call from Listen to Active
A Connect request is issued to the ISDN exchange
244
The Call_Manager registers an Observer for the call and adds the call to an array of calls
Call object is now an observer and in the array of call objects. Access "main" department table and embed in call object by invoking dept()
A Call object exists with an embedded "main" table
246
Is incoming call a voice call?
No
D
Yes
248
Call object plays the audio file to the caller
Call is actively coupled to a B-channel and shares a D-channel. The play() method of the call object is invoked to play the main voice menu
A

FIG. 10A

FIG. 10B


A
250
Call_Manager enables DTMF detection for the Call object
Call Manager invokes the EnableDTMF() method of the Call object
252
Caller makes a selection by pressing a key on his telephone keypad
DTMF tones from caller's keypad are received via from the B-channel by a DTMF detector in the Call object. DTMF detector was enabled by EnableDTMF()
E
254
DTMF event is received by the Call object and the Call_Manager. Call object identifies the key pressed
Callback system causes hardware event to be observed by Call_Manager and Call Object
256
Call_Manager obtains key from the Call object and uses key as row index of department table
Call_Manager invokes the ReadDTMF() method of the Call object
258
Call_Manager obtains Agent field from department table
Call_Manager invokes the getNextAgent(dtmf method using the key dtmf as the argument
260
Is Agent field empty?
Yes
No
262
Call_Manager gets the Agent's Directory Number from the Directory # field in the table
Call_Manager invokes the getNextAgent(dtmf) method using dtmf as the argument
264
Call_Manager Transfer call to Agent
Call_Manager invokes Transfer() method. After transfer call leaves the call management system, i.e., call is directly connected to agent
C
266
Embed Agent receiving call and Agent directory number in call object
Call Manager invokes agent (Agent) and DNagent (DNagent) methods
268
Call_Manager disconnects from call
Call Manager invokes Disconnect() method
End

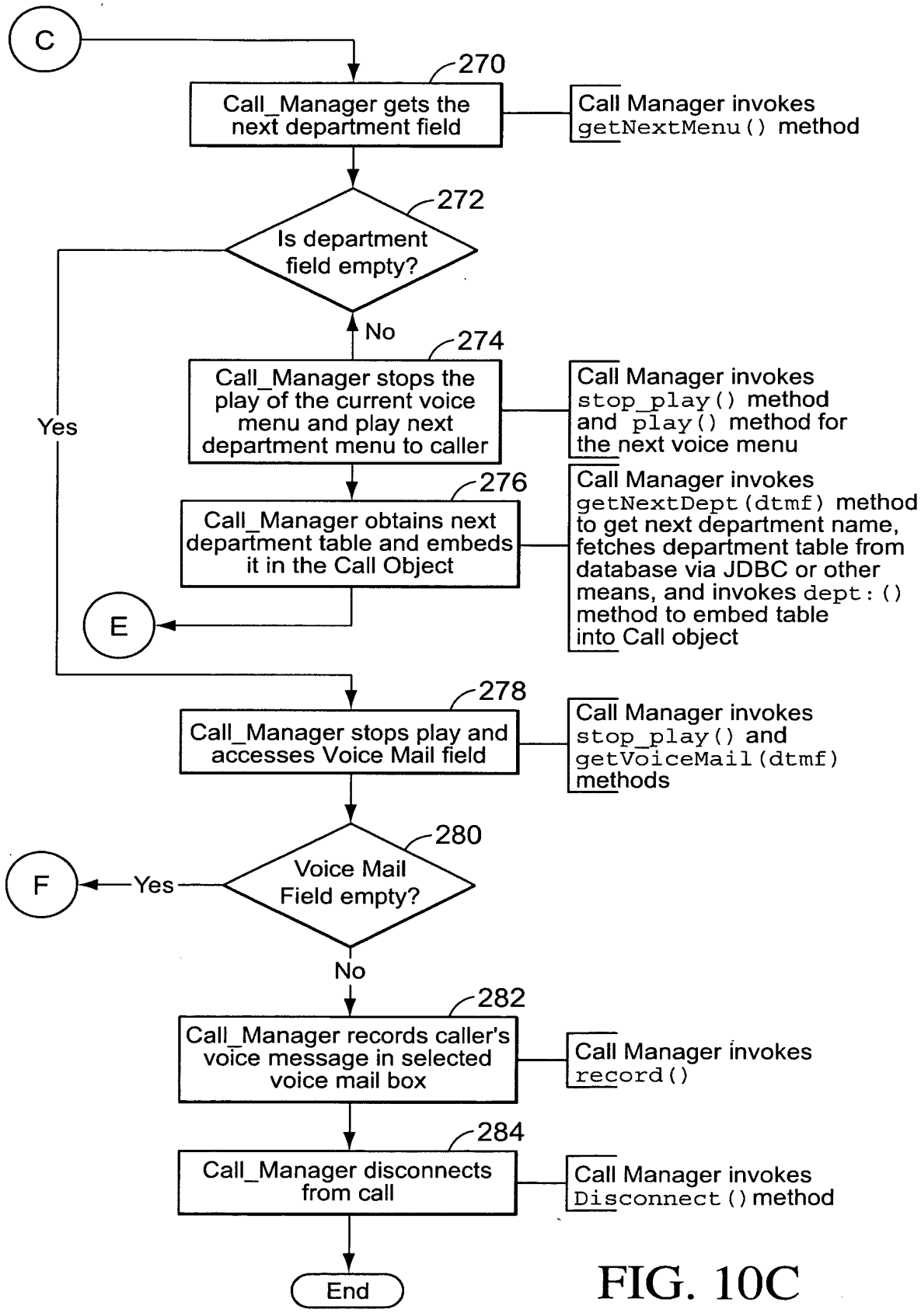

C
270
Call_Manager gets the next department field
Call Manager invokes getNextMenu() method
272
Is department field empty?
No
Yes
274
Call_Manager stops the play of the current voice menu and play next department menu to caller
Call Manager invokes stop_play() method and play() method for the next voice menu
276
Call_Manager obtains next department table and embeds it in the Call Object
Call Manager invokes getNextDept(dtmf) method to get next department name, fetches department table from database via JDBC or other means, and invokes dept:() method to embed table into Call object
E
278
Call_Manager stops play and accesses Voice Mail field
Call Manager invokes stop_play() and getVoiceMail(dtmf) methods
280
Voice Mail Field empty?
F
Yes
No
282
Call_Manager records caller's voice message in selected voice mail box
Call Manager invokes record()
284
Call_Manager disconnects from call
Call Manager invokes Disconnect() method
End

FIG. 10C

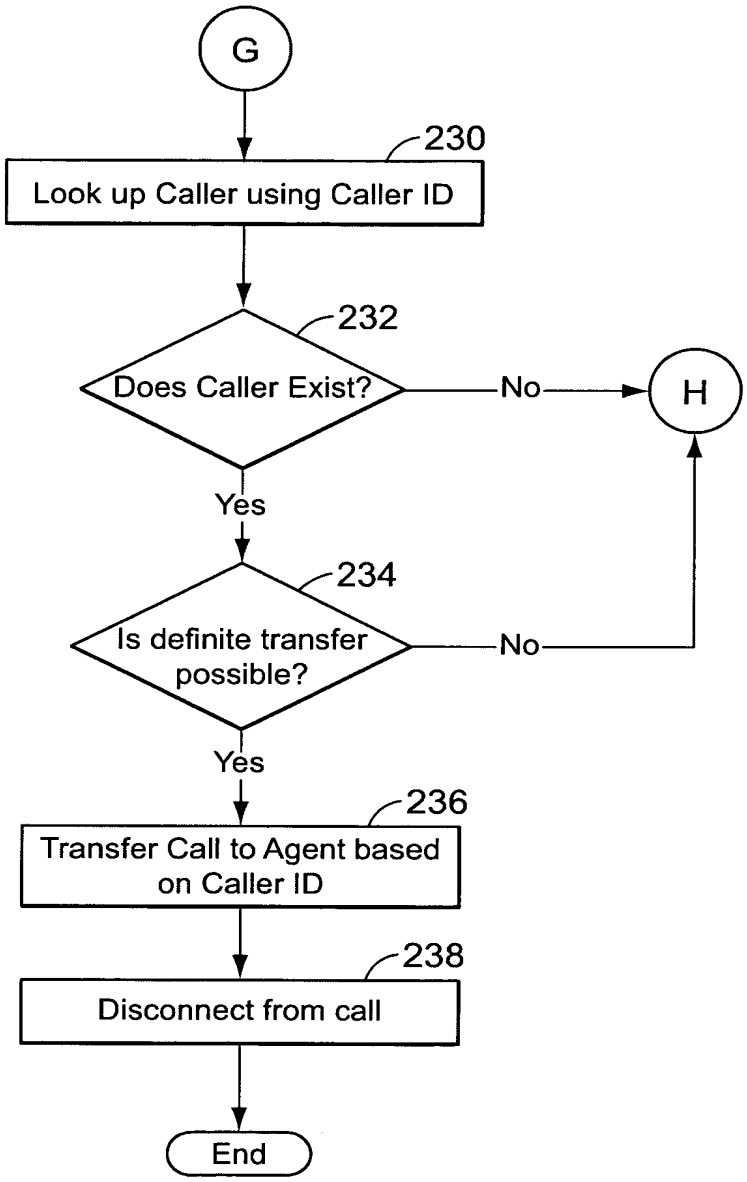

G
230
Look up Caller using Caller ID
232
Does Caller Exist?
No
H
Yes
234
Is definite transfer possible?
No
Yes
236
Transfer Call to Agent based on Caller ID
238
Disconnect from call
End

FIG. 10D

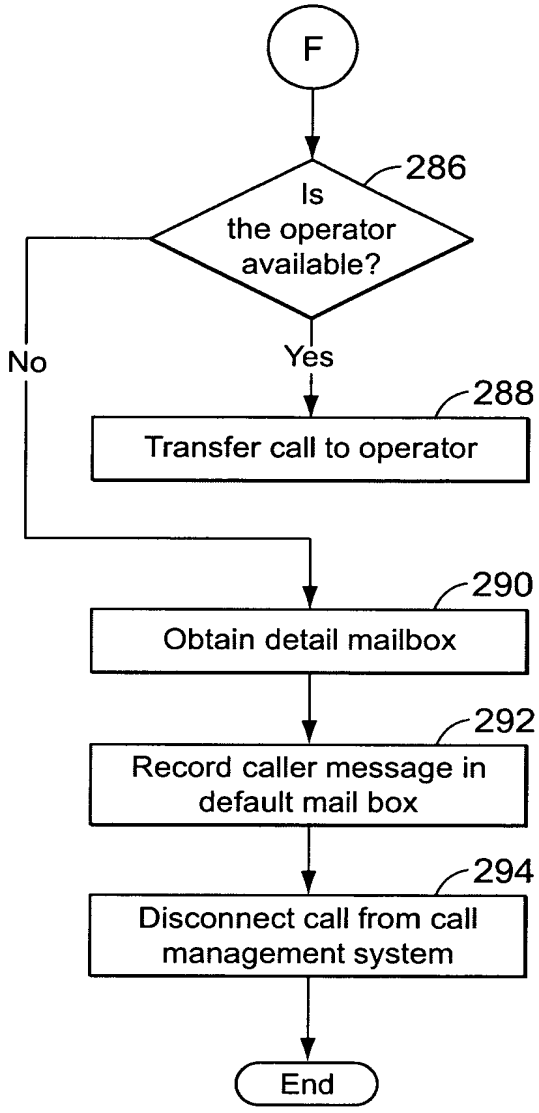

F
286
Is the operator available?
No
Yes
288
Transfer call to operator
290
Obtain detail mailbox
292
Record caller message in default mail box
294
Disconnect call from call management system
End

FIG. 10E

```
public void Listen()
{
    for( int n = 0; n < max, n++)
    {
        if ( Caller[n] == 0 )          // array of handles
        {
          Call[n]= new ISDNcall();

          Listener[n]= Call[n].Listen( Call_mask );
        }
    }
}
```

300

FIG. 11A

```
//|
//|  no agent avail, no next menu!
//|
        Call[n].stop_play();

        MailFile = thisDept.getVoiceMail( dtmd );

        saveLink = ISDNmenu.Open( MailFile );

        if( saveLink != null )
        {
          Call[n].record( saveLink );

        } else { // no agent, no menu, no mail
                 // switch to default voice mail
               }
 }
```

306

FIG. 11C

FIG. 11B

```
thisDept = Call[n].dept();
Agent = thisDept.getNextAgent( dtmf);
if( Agent != null )
  {
    // transfer call to Agent
        DNagent = thisDept.getAgentDN( dtmf );
        Xfer_ret = Call[n].Transfer();      // begin transfer
        if( Xfer_ret == 0 )      // success
        {
                Call[n].agent( Agent );
                Call[n].DNagent( DNagent );
        } else { // failed attempt to transfer call to Agent }
} else { // no Agent available or appropriate
```

302

```
// no Agent available or appropriate
        dtmenu = thisDept.getNextMenu( dtmf );
        Call[n].stop_play;      // stop playing voice menu
        if( dtmenu != null )
        {
        menuLink = ISDNmenu.open( dtmenu );
        if( menuLink != null )
          {
            Call[n].play( menuLink );
          } else {        // no menu avail  }
        } else {  // no voice menu avail  )
//|
//|  Now update the embedded departmant table:
//|
        Deptname = thisDept.getNextDept( dtmf );
  if( Deptname != null )
    {
        Department[n] = DeptTables.xlat( Deptname );
        //-- now embed next department model in Call[n]
        Call[n].dept( Tmodel{ Departmant[n] ]);
    }
  }
else { //no agent avail, no next menu!
```

304